Georgia Tech Psychology STING Telemetry Data Module For MiniSim

Georgia Tech School of Psychology Tech Report GT-PSYC-TR-2015-02

Thomas M. Gable, Bruce N. Walker, & Bhargav Rajendra

Thomas.Gable@gatech.edu, Bruce.Walker@psych.gatech.edu, bhargav.r@gatech.edu

August 27, 2015

Sonification Lab
School of Psychology
Georgia Institute of Technology
654 Cherry Street
Atlanta, GA, USA
http://sonify.psych.gatech.edu
Phone: +1-404-894-8265

0. Abstract

As in-vehicle technologies become more integrated into the vehicle, researchers must have real-time vehicle data available to them to investigate new technologies. In driving simulation settings, it can often be difficult to get this information. In an effort to gather real-time data from the National Advanced Driving Simulator MiniSim we have developed the Simulator Telemetry INteGration (STING) module to pull any data the MiniSim would normally output to its data acquisition files, and make those data available for use within other programs. The module is fairly simple to integrate into a MiniSim as it is purely software based. This document was written as a guide for those who are interested in using the STING software, to inform them of what STING can do, inform users of how the system works, and then linking them to the software download. Please refer to the included software license. If you use the system for studies please simply cite this tech report.

1. Overview of the Module

The Georgia Tech Simulator Telemetry INteGration (STING) software system provides an object-oriented approach to making data from the NADS MiniSim driving simulator available to third party modules or programs. STING exists as a MiniSim sub-module. It initializes a UDP server that can connect to multiple clients on different ports and transmit data in real time. The data are read prior to being sent to the MiniSim data acquisition files (DAQ files) but this does not interrupt the recording of data in MiniSim. This is a forking of the data path, instead of re-routing it. The module is built for use with MiniSim v2.

2. Use of the Module

In an effort to give other researchers this tool we are making the module open source for those using it for research and non-profit purposes. We appreciate suggestions and undertook this module with the assistance of engineers from the National Advanced Driving Simulator but this is not officially part of their software. Please also see the license that accompanies this software, especially with respect to the as-is nature of given source code. We recommend having this connection installed in a secondary (copied) folder directory, in a copy of the MiniSim software and its required files.

3. Requirements

To use this module, make sure to install the following software dependencies.

1) Microsoft .NET Framework v4
2) NADS MiniSim v2

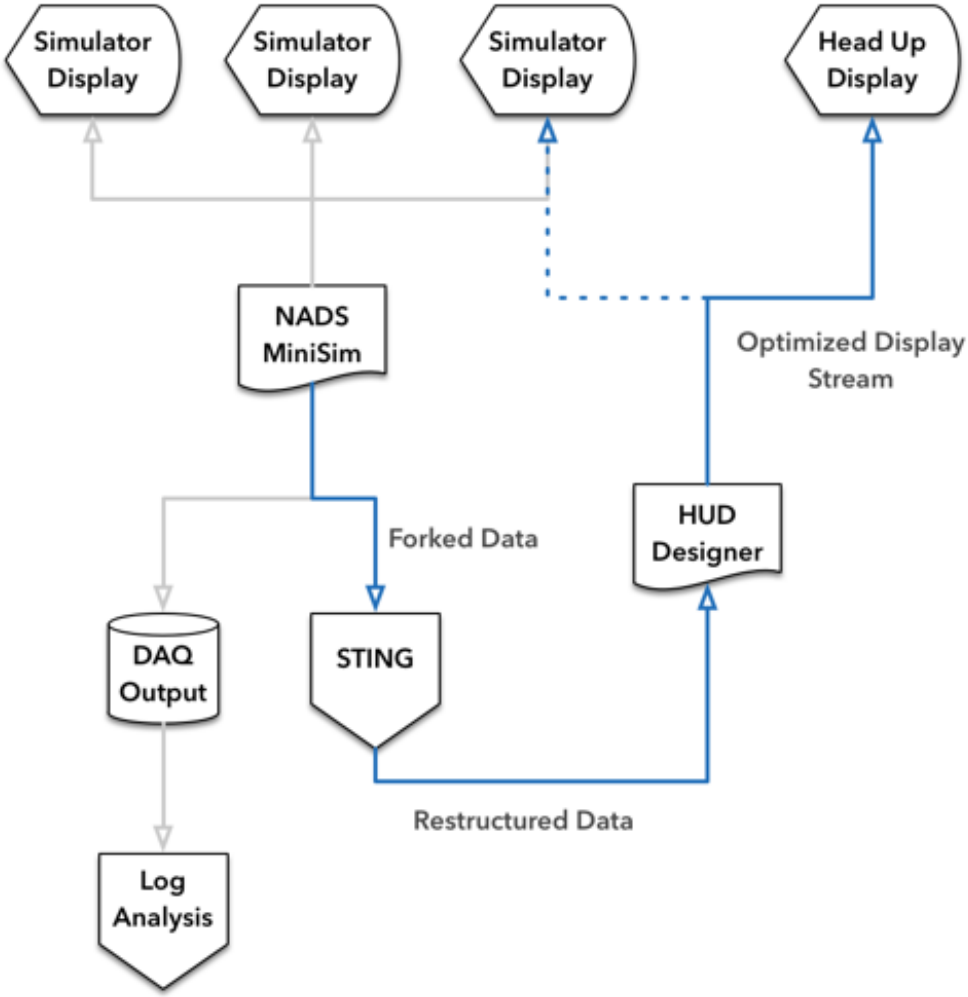
4. Dependencies

Modifying the STING software requires the following dependencies to be installed.

1) Microsoft Visual C++ 2010. Any version after 2010 modifies system headers and ends up breaking the system. Use the 2010 version.

2) Microsoft .NET Framework v4.

5. Architecture Diagram



6. Installing the STING Module

To install the STING software:

1) Check the code out from the repository, or download and unzip the package.
2) The binaries (EXE files) of the STING software need to be placed in bin folder of MiniSim.
3) Additionally, the STING software requires configuration files (CEC and DAQ.DF files). These files must also be placed in the bin folder of MiniSim.
4) MiniSim, on application load, also loads the STING server and information is ready to be consumed by any third party system.

7. Modifying the STING Module

To change what data variables are being output by the STING connection you must change the module code. To do this, follow the instructions below.

1) Check out or download code from the Georgia Tech SimulationServer repository.
2) Open the existing sln project in Visual C++ 2010. Make sure to use the 2010 version.
3) Open DaqMonitor.cpp in your project.
4) Register any new input elements you require from MiniSim. This example below shows how to register the current vehicle speed.

```
int speed_Status;
char speed_Msg[RTEX_MAX_ERR_STRING_SIZE];
if((speed_Status=RegInpElem("VDS_Veh_Speed",RTEX_FLOAT,&m_VDS_Veh_Speed,
sizeof(m_VDS_Veh_Speed)))< 0)
{
        sprintf(speed_Msg,"RegInpElem(VDS_Veh_Speed)
                failed: %s\r\n",ERR2Str(speed_Status));
        puts(speed_Msg);
        m_oAIF_Health_Status=0;
}
```

5) Override the sendMessage() method to provide a connection to your client machine.
6) The connection information required to communicate with a client are exposed as variables in DAQMonitor.cpp.
7) Add the sendMessage() snippet to the RunSubsytem() method of DAQMonitor.cpp under case MNADS_RUN
8) We're also providing examples of how to send data from the MiniSim, and receive the same data in real time from a sample client. Take a look at the MainWindow.xaml.cs file in "Sample UDP Data Sender" and "Sample UDP Data Receiver" to get started with data transmission.

   We use .NET Framework's sendTo() method to serve data on a predetermined port address over UDP. Similarly, we use .NET Framework's udpClient.Receive() method to read data from STING.

8. Questions and Contact

If there are questions or concerns about this system please contact the authors or Professor Bruce N. Walker at the Sonification Lab in the School of Psychology at the Georgia Institute of Technology. Other tech reports from the GT Sonification Lab also exist in the Georgia Tech SmartTech repository (https://smartech.gatech.edu/) regarding the MiniSim. This includes the Head Up Display (HUD) Designer program, which we developed to receive data from STING (or from other data sources) and then display animated or "live" gauges and displays in the MiniSim (GT Psychology Tech Report GT-PSYC-TR-2015-03) . We also have a MiniSim training module (GT Psychology Tech Report GT-PSYC-TR-2015-01), and a simulator sickness screening protocol (GT Psychology Tech Report GT-PSYC-TR-2013-01) if researchers are interested in any of those documents/resources.